





Research Article

Antenna Optimization Based on Auto-Context Broad Learning System

Wei-Tong Ding ¹, Fei Meng ², Yu-Bo Tian ² and Hui-Ning Yuan ¹

¹School of Electronics and Information, Jiangsu University of Science and Technology, Zhenjiang 212003, Jiangsu, China

²School of Information and Communication Engineering, Guangzhou Maritime University, Guangzhou 510725, Guangdong, China

Correspondence should be addressed to Yu-Bo Tian; tianyubo@just.edu.cn

Received 10 October 2021; Revised 11 January 2022; Accepted 29 January 2022; Published 21 February 2022

Academic Editor: Stefano Selleri

Copyright © 2022 Wei-Tong Ding et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

To enhance the efficiency of antenna optimization, surrogate model methods can usually be used to replace the full-wave electromagnetic simulation software. Broad learning system (BLS), as an emerging network with strong extraction ability and remarkable computational efficiency, has revolutionized the conventional artificial intelligence (AI) methods and overcome the shortcoming of excessive time-consuming training process in deep learning (DL). However, it is difficult to model the regression relationship between input and output variables in the electromagnetic field with the unsatisfactory fitting capability of the original BLS. In order to further improve the performance of the model and speed up the design of microwave components to achieve more accurate prediction of hard-to-measure quality variables through easy-to-measure parameter variables, the conception of auto-context (AC) for the regression scenario is proposed in this paper, using the current BLS training results as the prior knowledge, which are taken as the context information and combined with the original inputs as new inputs for further training. Based on the previous prediction results, AC learns an iterated low-level and context model and then iterates to approach the ground truth, which is very general and easy to implement. Three antenna examples, including rectangular microstrip antenna (RMSA), circular MSA (CMSA), and printed dipole antenna (PDA), and 10 UCI regression datasets are employed to verify the effectiveness of the proposed model.

1. Introduction

As is known to all, electromagnetic simulation software (EMSS) such as high-frequency structure simulator (HFSS) and computer simulation technology (CST) is most commonly used in the optimization design of electromagnetic devices, which can obtain high-precision results, however, along with high computational and time cost. Therefore, the use of surrogate models instead of EMSS for evaluating the fitness of electromagnetic components can save much optimization time, which is currently a hot topic in electromagnetic optimization design. Many popular modeling methods have been widely used like Gaussian process (GP) [1, 2], backpropagation (BP) [3], artificial neural network (ANN) [4–6], support vector machine (SVM) [7, 8], extreme learning machine (ELM) [9, 10], kernel ELM (KELM) [11],

and so on. Traditional deep neural networks are generally composed of multilayer learning to mine complex knowledge and abstract data characteristics from simple concepts, which have achieved breakthrough success [12], such as deep belief networks (DBN) [13] and convolutional neural networks (CNNs) [14]. Broad learning system (BLS) [15] is proposed to solve the problem of plenty of time and computing resources that the above-mentioned deep structure suffers due to the need to adjust large numbers of parameters and its complicated manual design structure. Subsequently, BLS was proven to have universal approximation capabilities [16]. Specifically, thanks to the fast incremental learning algorithm [17], which is applied to BLS, when faced with newly added samples and hidden nodes, the system can be updated incrementally without rebuilding the entire network from scratch. The construction of BLS is

based on the theory of random vector functional-link neural network (RVFLNN) [18, 19]; however, instead of directly bringing the original inputs into the network, BLS first maps them into feature nodes and then imitates the practice of RVFLNN, that is, nonlinearly transforming them into enhancement nodes, and these two parts together constitute the hidden layer of BLS. More importantly, the powerful mechanism for randomly generating hidden-layer node weights based on any continuous probability distribution is retained by BLS, so that only the output weights need to be trained through the pseudoinverse algorithm [20]. In particular, the input weights are first random generated and then fine-tuned by a sparse autoencoder. The fundamental characteristic of BLS is that the limit formula of Moore-inverse is utilized, and the pseudoinverse incremental formula is adopted, which can guarantee its training accuracy and fast incremental learning ability. Inspired by boosted neural nets [21] and a method of model fusion, that is, stacking, master-apprentice BLS (MABLS) [22] was proposed and applied to the antenna optimization. In [23], the probability that the current classification result belongs to each category is used as the context and combined with the original inputs as new inputs. Furthermore, Stacked BLS [24], which not only uses the current BLS's outputs, but also utilizes the training algorithm along with residual characteristic, was achieved by stacking several BLS blocks to approach the residual outputs of each block. In addition, a novel k-means clustering algorithm [25] based on a noise algorithm is developed, which solves the problem of determining the number of clusters and sensitively initializing the center cluster in the traditional k-means clustering algorithm. It is an iterative clustering analysis algorithm, which starts the next iteration according to the current clustering results. Still, coupled multistable stochastic resonance (CSMR) [26], adaptively optimizes and determines the system parameters of SR by using the output signal-to-noise ratio and seeker optimization algorithm and then feeds the preprocessed signal into CSMR for further training. Illuminated by these above approaches, we further propose auto-context BLS (ACBLS) as another version of MABLS, and the context is defined as the predicted values of the current model in regression problem.

The rest of the paper is organized as follows. Section 2 briefly reviews related works on BLS. In Section 3, the original BLS will be presented, and then the specific structure and algorithm of the proposed ACBLS are introduced in detail. Experiments on RMSA, CMSA, PDA, and 10 UCI regression data sets are conducted in Section 4 to demonstrate the proposed method and report the results and analysis. At last, Section 5 draws the conclusion.

2. Literature Review

Thanks to its extraordinary efficiency, prominent generalization performance, and easy extendibility, BLS has been applied in different domains. Due to space constraints, we will only show a portion of the innovations and applications on BLS. By incorporating TS fuzzy systems into a BLS, fuzzy BLS [27] is proposed, the feature

nodes are replaced by each group of Takagi–Sugeno fuzzy subsystems, and the input data are processed by each of them, fuzzy BLS retains the fast computational nature of BLS and can achieve great accuracies. A novel deep-broad learning system [28] is proposed to jointly consider effectiveness and efficiency in 5G era; specifically, based on typical BLS, it adopted long short-term memory to extract the mapped features, which further improve the performance of prediction. Xu et al. [29] propose a new recursive BLS to capture the dynamic nature of time series in order to make the network remember historical information, and the enhancement nodes are connected recursively. To model uncertain data, especially those with noise and outliers, Chen et al. [30] proposed robust BLS based on regularization and achieved great generalization. In order to fulfill the task of semisupervised classification, Zhao et al. [31] extended BLS based on popular regularization frameworks, forming a semisupervised BLS. However, so far, there are few applications in the field of electromagnetism. Until very recently, MABLS is applied for antenna optimization, and this paper is a continuation of this research.

3. The Proposed Algorithm Model

In this section, we will first briefly introduce the training process of standard BLS and then propose the detailed modeling strategy of ACBLS.

3.1. Standard Broad Learning System. BLS is a typical forward neural network, whose structure draws on the concept of RVFLNN and is divided into three layers: the input layer, the hidden layer, and the output layer. Constructing a BLS contains two essential procedures: (1) randomly generate the weights of both mapping feature nodes and enhancement nodes, and (2) calculate the weights between the hidden and output layers. The architecture of BLS is shown in Figure 1.

First of all, the inputs are converted into n sets of random feature nodes using mapping η_i , which is normally a linear mapping. The output of the i th group of mapping feature nodes can be denoted by the equation of the form

$$\mathbf{F}_i = \eta_i(\mathbf{X}\mathbf{W}_{ei} + \boldsymbol{\beta}_{ei}), \quad i = 1, 2, \dots, n, \quad (1)$$

where weights \mathbf{W}_{ei} and bias terms $\boldsymbol{\beta}_{ei}$ are randomly generated with proper dimensions. In particular, \mathbf{W}_{ei} is a matrix, which is then learned using a sparse autoencoder based on lasso regression.

Next, combining all the feature nodes, we have $\mathbf{F}^n \equiv [\mathbf{F}_1, \mathbf{F}_2, \dots, \mathbf{F}_n]$. Similarly, the enhancement layer $\mathbf{E}^m \equiv [\mathbf{E}_1, \mathbf{E}_2, \dots, \mathbf{E}_m]$ composed of m groups of enhancement nodes is obtained by transforming the feature layer \mathbf{F}^n using a nonlinear function ε_j and the j th group of enhancement nodes can be represented as

$$\mathbf{E}_j = \varepsilon_j(\mathbf{F}^n \mathbf{W}_{hj} + \boldsymbol{\beta}_{hj}), \quad j = 1, 2, \dots, m. \quad (2)$$

Once more, \mathbf{W}_{hj} and $\boldsymbol{\beta}_{hj}$ are the randomly generated matrix weights and bias terms. The hidden layer is composed

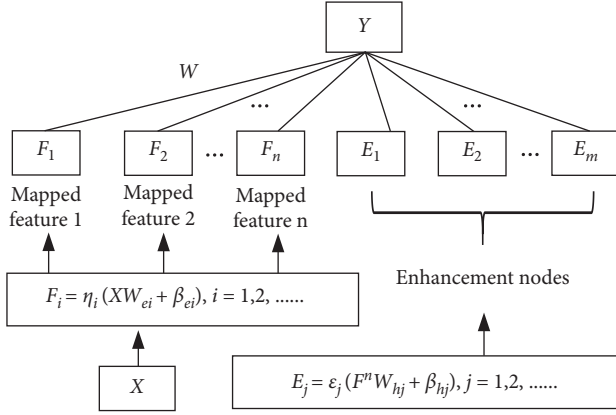


FIGURE 1: Structure of BLS

of the feature layer and the enhancement layer, which is expressed as

$$\mathbf{H} = [\mathbf{F}^n | \mathbf{E}^m]. \quad (3)$$

Hence, we have the outputs of the model

$$\mathbf{Y} = \mathbf{H}\mathbf{W}, \quad (4)$$

where \mathbf{W} are the weights from the hidden layer to the output layer and can be approximated rapidly by the ridge regression learning algorithm, shown in (5), which is a L_2 norm regularized least square problem, also referred to as the ridge regression problem.

$$\arg \min_{\mathbf{W}} : \|\mathbf{H}\mathbf{W} - \mathbf{Y}\|_2^2 + \frac{\lambda}{2} \|\mathbf{W}\|_2^2. \quad (5)$$

A constraint term λ , also called the regularization coefficient, is added to the original least squares estimate to make it possible to find the pseudoinverse when the original generalized inverse is under the ill condition. The solution of the above problem can be approximated as

$$\mathbf{W} = (\mathbf{H}^T \mathbf{H} + \lambda \mathbf{I})^{-1} \mathbf{H}^T \mathbf{Y}, \quad (6)$$

where \mathbf{H}^T is the transposed matrix of \mathbf{H} , and \mathbf{I} is the identity matrix. The solution verges to 0 when $\lambda \rightarrow \infty$. Specially, if $\lambda = 0$, the problem degenerates into the least square problem, and it is easy to obtain the solution of the original pseudoinverse.

3.2. Auto-Context Broad Learning System. Given a set of samples, the model is first trained to get its corresponding predictions, which are then used as the context information, that is, the prior knowledge, to train a new model. This procedure is somewhat similar to the part of the training process of three common integrated learning frameworks, which are Bagging, Boosting, and Stacking. For stacking integration, the predictors of a specified layer are independent of each other, so they can be trained in parallel on multiple servers. However, the predictor of a certain layer can only start training after all the predictors of the previous layer have been trained. Regardless of whether it is AC or

MA [22], each predictor is based on its previous results, and therefore, the training process must be orderly. It is meaningless to distribute it on multiple servers, and this feature is the same as boosting integration. The training process of AC can be summarized as follows:

- (1) When given a set of datasets together with their labels, $\mathbf{S} = \{\mathbf{X}, \mathbf{Y}\}$, $\mathbf{X} \in \mathbb{R}^{N \times M}$, $\mathbf{Y} \in \mathbb{R}^{N \times C}$, define \mathbf{Y}_u as the corresponding outputs of model- u , where u is the iteration index, $u = 1, 2, 3, \dots, U$.
- (2) Combine the context as each iteration's inputs $\mathbf{X}_u = [\mathbf{X} | \mathbf{Y}_{u-1}]$, $\mathbf{X}_u \in \mathbb{R}^{N \times (M+C)}$. It is worth noting that \mathbf{Y}_0 is the null matrix.
- (3) Use the model to calculate new outputs \mathbf{Y}_u based on new inputs \mathbf{X}_u .

The u -th AC iteration represents that the model- u will teach its prediction results to the model $u+1$; that is, we have to first complete the training of the original BLS and get the initial \mathbf{Y}_1 , and then we can start the AC iteration. Once a BLS has completed training, the algorithm repeats the same procedure to better approximate the ground truth. Without loss of generality, taking the k th teaching as an example, the structure of the proposed ACBLS is shown in Figure 2 and its training steps are presented in detail in Algorithm 1.

For most traditional ANNs, their structure is fixed, and the parameters need to be adjusted repeatedly in order to optimize the performance of the model. On the contrary, the parameters of BLS are randomly selected and fixed, and horizontal expansion is required to adjust the optimal structure. The above-mentioned unique feature is enabled no matter whether AC or MA is applied to the BLS, it could result in a fantastic performance, and there is no need to manually adjust any parameters after the training starts.

4. Experiments

In the first place, define the structure of the hidden layer of BLS as (F_n, M_g, E_n) , which are the numbers of feature nodes F_n , mapping groups M_g and enhancement nodes E_n , respectively. For MABLS, the hidden layer extends column C to the right, which contains the predictions of the previous model, so the final structure becomes (F_n, M_g, E_n, C) and the parameter C is conclusive and unchanging, which is the dimension of the model's output. So, whether the original BLS or the improved version using AC or MA, we only need to confirm (F_n, M_g, E_n) . The regularization parameter δ in ridge regression is set as 2^{-30} . It is worth mentioning that, except the output weights calculated by the ridge regression, all other weights and biases involved are randomly generated, which are drawn from the standard uniform distributions on the interval $[-1, 1]$. In particular, the input weights are fine-tuned by the sparse encoder with lasso regression of the input data to obtain better feature nodes, while the enhancement nodes are activated by the nonlinear activation function, which is the hyperbolic tangent function. All experiments are conducted on a computer equipped

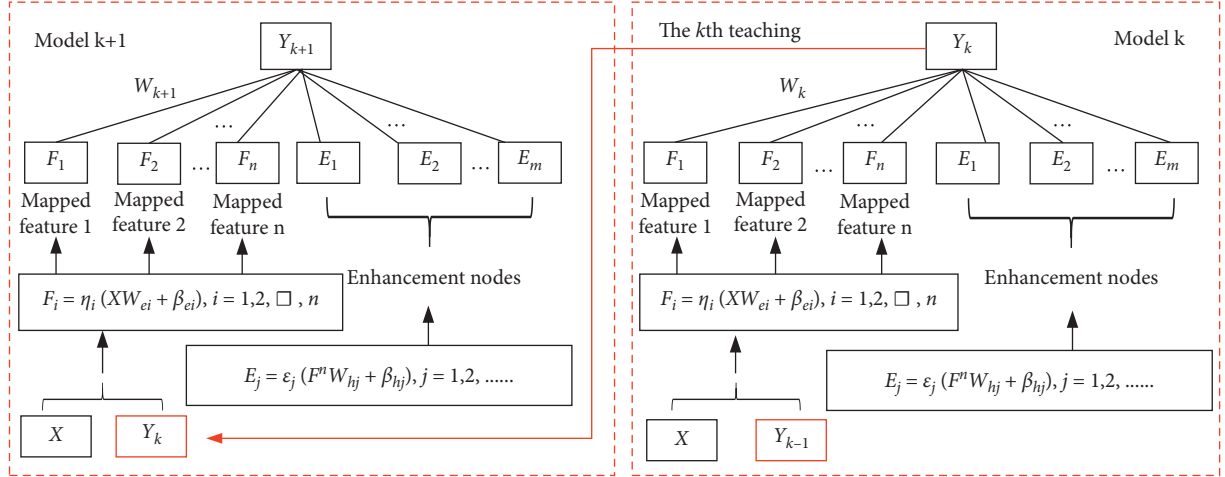


FIGURE 2: Structure of ACBLS

with Intel(R) Core(TM) i7-4790K CPU @ 4.00 GHz 4.00 GHz, and the RAM is 16 GB.

4.1. Resonant Frequency of RMSA. Taking the antenna of rectangular microstrip antenna (RMSA) [32, 33] as the first example. Figure 3 shows its top view schematic (above) and the side view (below). For this case, $\mathbf{X} \in \mathbb{R}^{33 \times 4}$ and $\mathbf{Y} \in \mathbb{R}^{33 \times 1}$, 26 samples are selected for training, and the remaining 7 groups are marked with asterisk * as test samples. All experimental data can be checked in [22] and will not be repeated here. Average percentage error (APE) given by (7) is utilized as the performance evaluation index to estimate the prediction errors of different modeling methods.

$$\text{APE} = \frac{1}{N} \sum_{i=1}^N \frac{|Y_i - y_i|}{|y_i|} \times 100, \quad (7)$$

where Y_i and y_i are the predicted value and the actual value, respectively. Besides, N is the number of samples.

In order to demonstrate the performance of the proposed ACBLS, 6 different modeling methods are compared to do the same experiment, including backpropagation (BP) [3], Parallel Tabu Search (PTS) [34], NN ensemble based on binary particle swarm optimization (BiPSO-NNE) [35], GP with ARD Matern 5/2 kernel function (GP52) [36], DKL with ARD Matern 5/2 kernel function (DKL52) [36], and MABLS [22].

For a fair comparison, just as the MABLS [22], we also perform a grid search from $[1,30] \times [1,30] \times [1,30]$ to determine the best structure (F_n, M_g, E_n) , the searching step is set to 1, and 8 iterations are taken. The optimal structure and test results of MABLS and ACBLS for each iteration are reported in Tables 1 and 2, respectively. It is worth noting that the k th iteration generates model $k+1$, and the first row of number 0 means that the current model is the original BLS, which is actually the model 1. To see the effects of iteration more clearly, the rows of Effect-1 and Effect-2 are added to make clear the degree to which each iteration is optimized over the previous result and the degree to which it is improved over the original one, respectively.

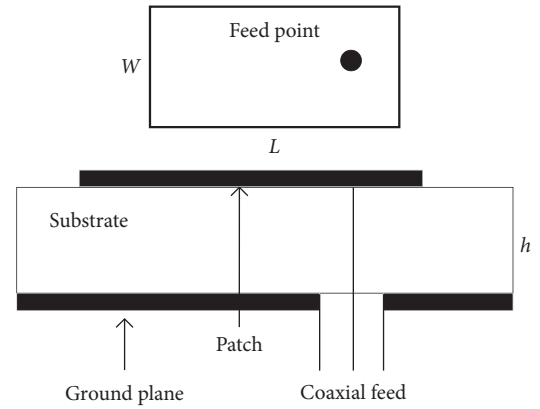


FIGURE 3: Schematic diagram of RMSA.

As observed in Table 1, the iterative results of MA show a decreasing trend, and it is a coincidence that AC in Table 2 shows the opposite phenomenon. However, after 8 iterations, the effect of MA is an astonishing 76.934%, while the effect of AC undergoing 8 iterations is only 48.997%, which is not as good as the effect of AC after the first iteration. We can preliminarily judge that the method of MA is more suitable than AC for the resonant frequency modeling of RMSA. In addition, the optimal test results of all the methods compared are presented in Table 3. After calculation, the prediction ability of ACBLS is 92.656%, 91.437%, 85.365%, 77.75%, 72.563%, and 48.997% higher than that of BP, PTS, BiPSO+NNE, GP52, DKL52, and BLS, respectively. For this case, the method of MA is 54.775% better than AC.

4.2. Resonant Frequency of CMSA. The second example is circular microstrip antenna (CMSA) [37, 38]; here are the relevant parameters: a is the radius of the circular patch, h is the thickness of the substrate, and ϵ_r is the relative dielectric constant. The above three parameters are used as inputs, while the outputs are the corresponding resonant frequency f . The top view schematic (above) and its side view (below)

Input: Training samples X ; Labels Y ;
Output: Y_{k+1}

- (1) **for** $u = 1$; $u \leq k + 1$
- (2) Combine the context as model u 's inputs $\mathbf{X}_u = [\mathbf{X}|\mathbf{Y}_{u-1}]$, where \mathbf{Y}_0 is the null matrix.
- (2) **for** $i = 1$; $i \leq n$ **do**
- (3) Random $\mathbf{W}_{ei}, \beta_{ei}$
- (4) Calculate $\mathbf{F}_i = \eta_i(\mathbf{X}_u \mathbf{W}_{ei} + \beta_{ei}), i = 1, 2, \dots, n$
- (5) **end**
- (6) Set the feature group $\mathbf{F}^n \equiv [\mathbf{F}_1, \mathbf{F}_2, \dots, \mathbf{F}_n]$
- (7) **for** $j = 1$; $j \leq m$ **do**
- (8) Random $\mathbf{W}_{hj}, \beta_{hj}$
- (9) Calculate $\mathbf{E}_j = \varepsilon_j(\mathbf{F}^n \mathbf{W}_{hj} + \beta_{hj}), j = 1, 2, \dots, m$
- (10) **end**
- (11) Set the enhancement group $\mathbf{E}^m \equiv [\mathbf{E}_1, \mathbf{E}_2, \dots, \mathbf{E}_m]$
- (12) Set original hidden layer of model u $\mathbf{H}_u = [\mathbf{F}^n|\mathbf{E}^m]$
- (13) Calculate the output weights of model u $\mathbf{W}_u = (\mathbf{H}_u^T \mathbf{H}_u + \lambda \mathbf{I})^{-1} \mathbf{H}_u^T \mathbf{Y}$
- (14) Calculate and save the outputs of model u $\mathbf{Y}_u = \mathbf{H}_u \mathbf{W}_u$
- (15) **end**

ALGORITHM 1: ACBLS – Auto-context of k iterations.

TABLE 1: MA Iteration results on RMSA.

Iteration	0	1	2	3	4	5	6	7	8
Structure	(7,8,15)	(9,19,27)	(7,10,9)	(12,30,3)	(15,23,2)	(30,27,2)	(15,22,1)	(16,20,1)	(26,16,1)
APE	0.2792	0.1113	0.0745	0.0683	0.0657	0.0651	0.0647	0.0645	0.0644
Effect-1	—	60.136%	33.064%	8.322%	3.807%	0.913%	0.614%	0.309%	0.155%
Effect-2	—	60.136%	73.317%	75.537%	76.468%	76.683%	76.827%	76.898%	76.934%

TABLE 2: AC iteration results on RMSA.

Iteration	0	1	2	3	4	5	6	7	8
Structure	(7,8,15)	(14,3,3)	(25,18,4)	(6,20,2)	(29,15,1)	(27,28,15)	(18,16,5)	(24,27,7)	(20,13,3)
APE	0.2792	0.213	0.1994	0.1962	0.1953	0.1738	0.1531	0.1441	0.1424
Effect-1	—	23.711%	6.385%	1.605%	0.459%	11.009%	11.91%	5.879%	1.388%
Effect-2	—	23.711%	28.582%	29.728%	30.05%	37.751%	45.165%	48.388%	48.997%

TABLE 3: Results of different models on RMSA.

f/Mhz	BP	PTS	BiPSO + NNE	GP52	DKL52	BLS	ACBLS	MABLS
3200	3178	3167	3196	3196.581	3196.579	3198.6	3200.1	3200
3580	3644.6	3685.2	3614	3582.141	3584.779	3589.2	3580.8	3580
4805	4703.3	4879	4796	4830.272	4823.193	4835.7	4818	4805.1
5100	5291.4	5191.4	5182	5145.394	5088.715	5084.1	5088.5	5099.7
5820	5924.5	5780.3	5853	5850.37	5835.184	5841.1	5847.3	5837.1
6200	6147.2	6205.7	6196	6198.311	6197.412	6204.4	6200.4	6208.8
8450	8233.1	8148.6	8169	8251.843	8239.441	8427.1	8450.2	8449.4
APE	1.939	1.663	0.973	0.640	0.519	0.2792	0.1424	0.0644

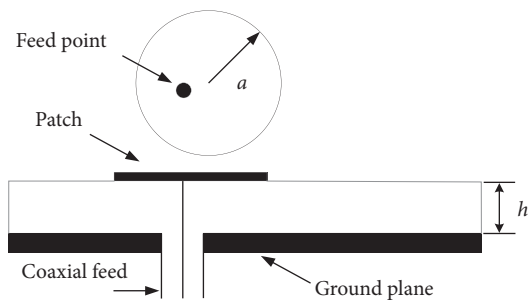


FIGURE 4: Schematic diagram of CMSA.

are shown in Figure 4. For this case, $X \in \mathbb{R}^{20 \times 3}$ and $Y \in \mathbb{R}^{20 \times 1}$. All experimental data are tabulated in Table 4, among which 16 of them are selected as training samples, and the remaining 4 groups are marked with asterisk * as testing samples.

Six different modeling methods with the same experiment are compared to validate the prediction precision of the proposed ACBLS, including BiPSO-NNE, delta-bar-delta (DBD) [3], BP, PTS, extended DBD (EDBD) [3], and MABLS. For a fair comparison, for BLS, MABLS, and ACBLS, we perform the same grid search

TABLE 4: Resonant frequency of CMSA for TM11 mode.

No	a/cm	h/cm	ϵ_r	f/MHz
1	6.8	0.08	2.32	835
2 *	6.8	0.159	2.32	829
3	6.8	0.318	2.32	815
4	5	0.159	2.32	1128
5	3.8	0.1524	2.49	1443
6	4.85	0.318	2.52	1099
7 *	3.493	0.1588	2.5	1570
8	1.27	0.0794	2.59	4070
9	3.493	0.3175	2.5	1510
10	4.95	0.235	4.55	825
11	3.975	0.235	4.55	1030
12	2.99	0.235	4.55	1360
13 *	2	0.235	4.55	2003
14	1.04	0.235	4.55	3750
15	0.77	0.235	4.55	4945
16	1.15	0.15875	2.65	4425
17	1.07	0.15875	2.65	4723
18	0.96	0.15875	2.65	5524
19 *	0.74	0.15875	2.65	6634
20	0.82	0.15875	2.65	6074

from $[1,30] \times [1,30] \times [1,15]$ to determine the best structure (F_n, M_g, E_n) , the searching step is set to 1, and 8 iterations are taken.

The optimal structure and testing results of MABLS and ACBLS for each iteration are listed in Tables 5 and 6, respectively. It can be readily seen from Tables 5 and 6 that both AC and MA have great performance in the prediction of the resonance frequency of CMSA. Moreover, results of the first 4 iterations of MA are better than those of AC; however, the effect of subsequent iterations for AC is more obvious and has always been at an advantage. Table 7 gives the best prediction results of different methods, and after calculation, the performance of ACBLS is 99.978%, 99.963%, 99.951%, 99.931%, 99.898%, and 99.843% higher than that of BiPSO + NNE, DBD, BP, PTS, BLS, EDBD, and BLS, respectively. For this case, the method of AC is 76.25% better than MA.

4.3. Printed Dipole Antenna. The top view of the printed dipole antenna (PDA) is shown in Figure 5 and the corresponding three-dimensional view in HFSS is presented in Figure 6. The design index of PDA is to operate at the working frequency of 2.45 GHz. The structure of the antenna can be divided into five parts, which are the dielectric layer, the dipole antenna arm, the microstrip Barron line, the microstrip transmission line, and the feed surface. Five influential geometrical variables each with five levels, i.e., $\mathbf{X} = [L1, L2, L3, L4, W3]$, are the inputs of the model, which are transmission line length $L1$, dipole arm length $L2$, Barron triangle side right angle side length $L3$, Barron triangle base right angle side length $L4$, and microwave Barron rectangle section width $W3$, respectively. The value ranges of parameters and sampling intervals are defined in Table 8. The unit is millimeter, and other parameters are fixed values, including the thickness of the dielectric layer $H=1.6$ mm, transmission line width $W1=3$ mm, and dipole sheet width $W2=3$ mm. In addition, the relative dielectric constant ϵ_r is 4.4.

Since return loss (S_{11}) is one of the important indicators for analyzing the performance of antenna, this paper verifies the effectiveness of the proposed algorithm by fitting the curve of S_{11} . The frequency scanning range of PDA is set to 2 GHz~3 GHz with a step size of 0.001 GHz, which means that each set of the inputs corresponds to 1001 outputs. Thus, for this case, $\mathbf{X} \in \mathbb{R}^{31 \times 5}$ and $\mathbf{Y} \in \mathbb{R}^{31 \times 1001}$.

By using the HFSS-MATLAB-API script [39], HFSS software is called by the scripts programmed in MATLAB to get the outputs, which is the S_{11} . Through partial orthogonal experiments, 31 samples are generated, 25 of which are used as training sets, and the rest are used as test sets. APE given by the above (7) is used as the performance evaluation index. We perform a grid search for the original BLS from $[1,10] \times [1,10] \times [1,70]$ to determine the optimal structure (F_n, M_g, E_n) , and the searching step is set to 1. The modeling time of BLS is 111.87 s, however, considering that AC needs iteration if, in order to further save the training time, when executing AC on BLS, the scope of grid search can be narrowed down to $[1,10] \times [1,10] \times [1,10]$, and the training time of each AC is only 26.33 s. Clearly, the total training time of ACBLS is the sum of the training time spent by the original BLS and the time required for K iterations of AC, and the optimization times of related methods are listed in Table 9.

The results of AC iteration are all recorded in Table 10 and after 8 iterations, the effect of the original BLS is optimized by 40.029%. For this case, the iteration is chosen as 8, the training time here is 111.87s(BLS) +8* 26.33s(8 AC iterations), which is only 322.51s, compared with the direct EM simulation for optimization, and the proposed method takes much less CPU time. One set of obtained optimal S_{11} solution satisfying the antenna criterion is plotted in Figure 7, and the corresponding geometric value is $\mathbf{X} = [22, 21, 10, 12, 3]$ mm. Meanwhile, the blue line of "HFSS" is the simulation result, and the red line of "proposed" is the prediction result. The S_{11} reaches -23.991 dB @2.45 GHz,

TABLE 5: MA iteration results on CMSA.

Iteration	0	1	2	3	4	5	6	7	8
Structure	(17,27,13)	(27,21,12)	(20,15,5)	(17,19,5)	(18,14,1)	(28,23,1)	(24,26,2)	(23,5,1)	(19,4,1)
APE	0.483	0.1344	0.0759	0.035	0.0109	0.0058	0.0044	0.0042	0.0032
Effect-1	—	72.174%	43.527%	53.887%	68.857%	46.789%	24.138%	4.545%	23.809%
Effect-2	—	72.174%	84.286%	92.754%	97.743%	98.799%	99.089%	99.130%	99.337%

TABLE 6: AC iteration results on CMSA.

Iteration	0	1	2	3	4	5	6	7	8
Structure	(17,27,13)	(8,22,4)	(7,13,1)	(19,18,12)	(15,28,5)	(25,7,6)	(10,2,3)	(6,4,4)	(16,1,1)
APE	0.483	0.154	0.1147	0.1016	0.0138	0.0031	0.0016	0.00098	0.00076
Effect-1	—	68.116%	25.519%	11.421%	86.417%	77.536%	48.387%	38.75%	22.449%
Effect-2	—	68.116%	76.253%	78.965%	97.143%	99.358%	99.669%	99.797%	99.843%

TABLE 7: Results of different models on CMSA.

f /Mhz	BiPSO + NNE	DBD	BP	PTS	EDBD	BLS	MABLS	ACBLS
829	776.3	792.4	817.4	850.4	820.2	831.22	828.968	829.0002
1570	1509.9	1509.9	1623.9	1566.9	1565.5	1559.9	1570	1570
2003	1952	1952	1975.4	2030.8	1979.1	2007.9	2003	2003
6634	6576.4	6576.4	6636.8	6616.4	6662.5	6685.5	6634.5	6634
APE	3.3999	2.0508	1.5631	1.108	0.7427	0.483	0.0032	0.00076

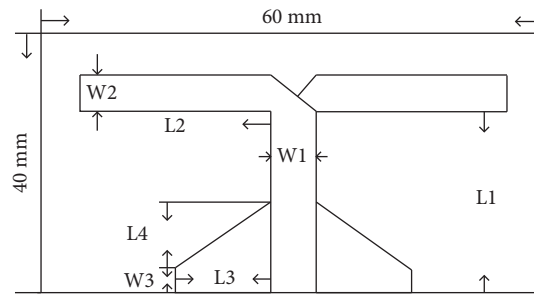


FIGURE 5: Structure diagram of the PDA.

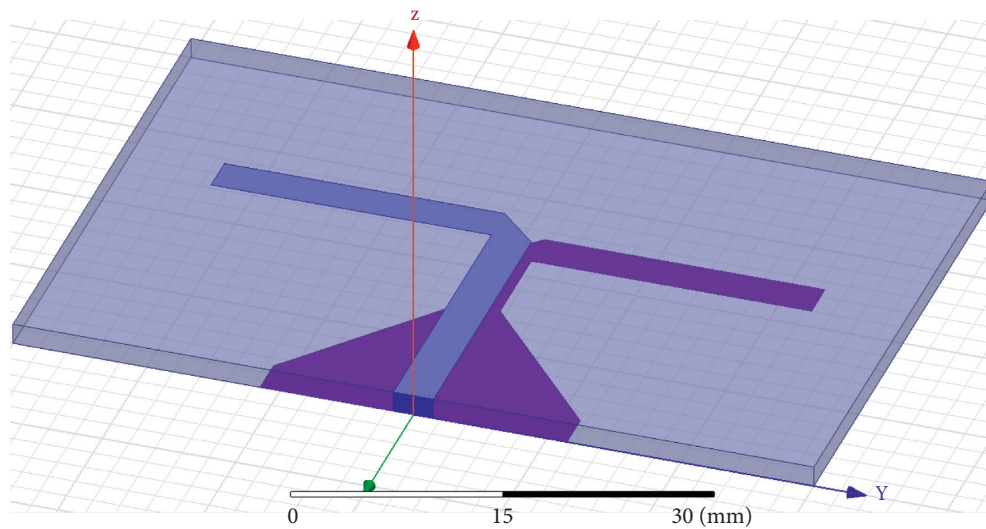


FIGURE 6: HFSS model of the PDA.

TABLE 8: Experimental samples of the PDA.

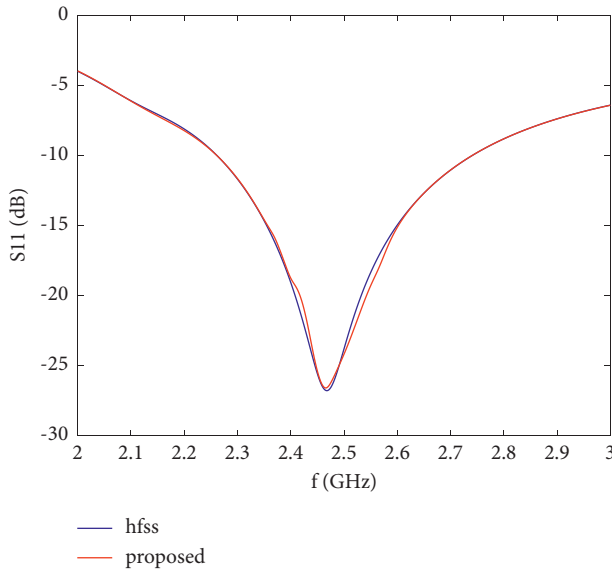
Variable	Min	Max	Step
$L1(\text{mm})$	21	23	0.5
$L2(\text{mm})$	20	22	0.5
$L3(\text{mm})$	9	10	0.25
$L4(\text{mm})$	11	12	0.25
$W3(\text{mm})$	2.7	3.3	0.15

TABLE 9: Optimization times of the relevant methods.

Method	Time/s
HFSS	1476.34
BLS	111.87
ACBLS	322.51

TABLE 10: AC iteration results on the PDA.

Iteration	0	1	2	3	4	5	6	7	8
Structure	(4,5,67)	(1,2,9)	(5,10,1)	(5,9,4)	(9,1,1)	(5,9,2)	(4,3,1)	(3,10,1)	(4,10,1)
APE	2.5786	2.2550	1.9040	1.7762	1.7586	1.6748	1.6545	1.6118	1.5464
Effect-1	—	12.549%	15.565%	6.712%	0.99%	4.765%	1.212%	2.581%	4.058%
Effect-2	—	12.549%	26.161%	31.118%	31.800%	35.050%	35.837%	37.493%	40.029%

FIGURE 7: $(S)_{11}$ fitting diagram of the PDA.

which meets the design requirements. It can be easily seen that the modeled and simulated results are highly consistent, which proves the validity of the proposed model.

4.4. Regression Data Sets. Considering some compared methods in the above experiment stage, which may not be commonly used and state-of-the-art methods, typical models such as SVM [8], LSSVM, ELM [40]₂ and the latest improved version of Greedy BLS (GBLS) [41] are involved in the comparison of 10 real-world regression data sets from the University of California, Irvine (UCI) database [42]₂ to further highlight the validity of the proposed algorithm, and the details of different data sets are put up in Table 11. For

TABLE 11: Details of data sets.

Data	Training number	Testing number	Input variables
Abalone	2784	1393	8
Basketball	64	32	4
Bodyfat	168	84	14
Cleveland	202	101	13
Housing	337	169	13
Mortgage	699	350	15
Pyrim	49	25	27
Quake	1452	726	3
Strike	416	209	6
Weather	974	487	9

fair comparisons, the same grid search is performed from $[1,10] \times [1,30] \times [1,200]$ for BLS, GBLS, MABLS, and ACBLS, and the searching step is set to 1.

As in the previous cases, 8 iterations are considered for MA and AC. Root mean square error (RMSE) [43] is selected as the performance index, and the optimal testing results of different models are tabulated in Table 12, of which the best RMSE result corresponding to each data set is indicated in bold.

$$\mathbf{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - Y_i)^2}, \quad (8)$$

where N is the number of samples, Y_i is the predicted value, and y_i is the actual value.

It can be readily seen from Table 12 that, under the same condition, the method of AC outperforms any model compared in all the experiments on 10 function approximation data sets except MABLS in the case of Pyrim. Ultimately, for Bodyfat, Housing, Strike, and Basketball, the performance of AC is better than MA by 10%, 9.259%, 4.267%, and 3.933%, respectively, and the improvement

TABLE 12: RMSE testing results for different models.

Data	SVM	LSSVM	ELM	BLS	GBLS	MABLS	ACBLS
Abalone	0.0773	0.0756	0.0777	0.0754	0.0747	0.0740	0.0734
Basketball	0.0767	0.0744	0.0719	0.0659	0.0628	0.0534	0.0513
Bodyfat	0.0049	0.0038	0.0033	0.0030	0.0027	0.0020	0.0018
Cleveland	0.1514	0.1256	0.1281	0.1199	0.1061	0.1039	0.1010
Housing	0.0792	0.0780	0.0762	0.0751	0.0711	0.0594	0.0539
Mortgage	0.0046	0.0053	0.0057	0.0043	0.0040	0.0036	0.0035
Pyrim	0.1549	0.1133	0.1376	0.0578	0.0381	0.0209	0.0301
Quake	0.2029	0.1733	0.1729	0.1718	0.1712	0.1711	0.1710
Strike	0.1053	0.1007	0.1043	0.1001	0.0964	0.0961	0.0920
Weather	0.0190	0.0191	0.0191	0.0188	0.0186	0.0180	0.0178

effect for the remaining data sets is less than 3%. When ACBLS is compared with the original BLS, for Pyrim, Bodyfat, Housing, and Basketball, the improvements are 47.924%, 40%, 28.229%, and 22.155%, respectively, and the remaining six data sets show improvements of less than 20%. It can be concluded that the predictive power of ACBLS is slightly better than that of MABLS in most cases, and the former performs much better than other above-mentioned models compared, and up to now, the effectiveness of the proposed model is further verified.

5. Conclusion

In this paper, we have developed and evaluated ACBLS, which obtains context features in the previous regression results. Our goal is to design an iterated framework to rapidly and effectively propagate and use the context information. It is very general and easy to implement and does not depend on any particular type of model, which can avoid heavy algorithm design such as various energy terms and procedures. Three antenna cases and 10 UCI regression datasets are illustrated and comparative results triumphantly demonstrate that the proposed method greatly improved the unsatisfactory generalization ability of the original BLS, and its modeling capability far exceeds that of some mainstream methods. It can be concluded that the proposed model may provide an efficient and powerful parametric modeling ability in antenna optimization, replacing the time-consuming EMSS.

Data Availability

The data that supported the findings of this study are available from the corresponding author upon request.

Disclosure

Weitong Ding and Fei Meng are co-first authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC) under No.61771225, and the scientific research capacity improvement project of key

developing disciplines in Guangdong Province of China under No. 2021ZDJS057.

References

- [1] J. P. Jacobs, "Efficient resonant frequency modeling for dual-band microstrip antennas by Gaussian process regression," *IEEE Antennas and Wireless Propagation Letters*, vol. 14, pp. 337–341, 2015.
- [2] M. I. Jais, T. Sabapathy, M. Jusoh et al., "A fuzzy-based angle-of-arrival estimation system (AES) using radiation pattern reconfigurable (RPR) antenna and modified Gaussian membership function," *IEEE Access*, vol. 7, pp. 145477–145488, 2019.
- [3] K. Guney, S. Sagioglu, and M. Erler, "Generalized neural method to determine resonant frequencies of various microstrip antennas," *International Journal of RF and Microwave Computer-Aided Engineering*, vol. 12, no. 1, pp. 131–139, 2002.
- [4] Y. Chen, Y. B. Tian, Z. Qiang, and L. Xu, "Optimisation of reflection coefficient of microstrip antennas based on KBNN exploiting GPR model," *IET Microwaves, Antennas & Propagation*, vol. 12, no. 4, pp. 602–606, 2018.
- [5] L.-Y. Xiao, W. Shao, X. Ding, Q. H. Liu, and W. T. Joines, "Multigrade Artificial neural network for the design of finite periodic arrays," *IEEE Transactions on Antennas and Propagation*, vol. 67, no. 5, pp. 3109–3116, 2019.
- [6] L.-Y. Xiao, W. Shao, F.-L. Jin, and B.-Z. Wang, "Multiparameter modeling with ANN for antenna design," *IEEE Transactions on Antennas and Propagation*, vol. 66, no. 7, pp. 3718–3723, 2018.
- [7] F. Y. Sun, Y. B. Tian, and Z. L. Ren, "Modeling the resonant frequency of compact microstrip antenna by the PSO-based SVM with the hybrid kernel function," *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, vol. 29, no. 6, pp. 1129–1139, 2016.
- [8] M. Tarkowski and L. Kulas, "RSS-based DoA estimation for ESPAR antennas using support vector machine," *IEEE Antennas and Wireless Propagation Letters*, vol. 18, no. 4, pp. 561–565, 2019.
- [9] Y. Gao and Y. B. Li, "Misfire identification of automobile engines based on wavelet packet and extreme learning machine," *Testing Science and Instruments*, vol. 4, pp. 384–395, 2017.
- [10] H. Zhang, B. Ai, W. Xu, L. Xu, and S. Cui, "Multi-antenna channel interpolation via tucker decomposed extreme learning machine," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 7, pp. 7160–7163, 2019.
- [11] L.-Y. Xiao, W. Shao, X. Ding, and B.-Z. Wang, "Dynamic adjustment kernel extreme learning machine for microwave

- component design," *IEEE Transactions on Microwave Theory and Techniques*, vol. 66, no. 10, pp. 4452–4461, 2018.
- [12] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [13] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.
- [14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, 2012.
- [15] C. L. P. Chen and Z. Liu, "Broad learning system: an effective and efficient incremental learning system without the need for deep architecture," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 1, pp. 10–24, 2018.
- [16] C. L. P. Chen, Z. Liu, and S. Feng, "Universal approximation capability of broad learning system and its structural variations," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 4, pp. 1191–1204, 2019.
- [17] C. L. P. Chen and J. Z. Wan, "A rapid learning and dynamic stepwise updating algorithm for flat neural networks and the application to time-series prediction," *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 29, no. 1, pp. 62–72, 1999.
- [18] Y.-H. Pao and Y. Takefuji, "Functional-link net computing: theory, system architecture, and functionalities," *Computer*, vol. 25, no. 5, pp. 76–79, 1992.
- [19] Y.-H. Pao, G.-H. Park, and D. J. Sobajic, "Learning and generalization characteristics of the random vector functional-link net," *Neurocomputing*, vol. 6, no. 2, pp. 163–180, 1994.
- [20] A. E. Hoerl and R. W. Kennard, "Ridge regression: biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.
- [21] S. Badirli, X. Liu, Z. Xing, A. Bhowmik, K. Doan, and S. S. Keerthi, "Gradient boosting neural networks: Grownnet," arXiv preprint arXiv:2002.07971, 2020.
- [22] W. T. Ding, Y. B. Tian, P. F. Li, H. Yuan, and R. Li, "Antenna optimization based on master-apprentice broad learning system," *International Journal of Machine Learning and Cybernetics*, vol. 13, no. 2, pp. 461–470, 2021.
- [23] Z. Xiang Bai and X. Bai, "Auto-context and its application to high-level vision tasks and 3D brain image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 10, pp. 1744–1757, 2010.
- [24] Z. Liu, C. L. P. Chen, S. Feng, Q. Feng, and T. Zhang, "Stacked broad learning system: from incremental flattened structure to deep model," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 1, pp. 209–222, 2021.
- [25] X. Ran, X. Zhou, M. Lei, W. Tepsan, and W. Deng, "A novel k-means clustering algorithm with a noise algorithm for capturing urban hotspots," *Applied Sciences*, vol. 11, no. 23, p. 11202, 2021.
- [26] H. Cui, Y. Guan, H. Chen, and W. Deng, "A novel advancing signal processing method based on coupled multi-stable stochastic resonance for fault detection," *Applied Sciences*, vol. 11, no. 12, p. 5385, 2021.
- [27] S. Feng and C. L. P. Chen, "Fuzzy broad learning system: a novel neuro-fuzzy model for regression and classification," *IEEE Transactions on Cybernetics*, vol. 50, no. 2, pp. 414–424, 2020.
- [28] M. Chen, X. Wei, Y. Gao, L. Huang, M. Chen, and B. Kang, "Deep-broad learning system for traffic flow prediction toward 5G cellular wireless network," in *Proceedings of the 2020 International Wireless Communications and Mobile Computing (IWCMC)*, pp. 940–945, IEEE, Limassol, Cyprus, July 2020.
- [29] M. Xu, M. Han, C. L. P. Chen, and T. Qiu, "Recurrent broad learning systems for time series prediction," *IEEE Transactions on Cybernetics*, vol. 50, no. 4, pp. 1405–1417, 2020.
- [30] J. Jin, C. L. P. Chen, and Y. Li, "Robust broad learning system for uncertain data modeling," in *Proceedings of the 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 3524–3529, Miyazaki, Japan, 2018.
- [31] H. Zhao, J. Zheng, W. Deng, and Y. Song, "Semi-supervised broad learning system based on manifold regularization and broad network," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 3, pp. 983–994, 2020.
- [32] M. Kara, "Closed-form expressions for the resonant frequency of rectangular microstrip antenna elements with thick substrates," *Microwave and Optical Technology Letters*, vol. 12, no. 3, pp. 131–136, 1996.
- [33] M. Kara, "The resonant frequency of rectangular microstrip antenna elements with various substrate thicknesses," *Microwave and Optical Technology Letters*, vol. 11, no. 2, pp. 55–59, 1996.
- [34] S. Sagiroglu and A. Kalinli, "Determining resonant frequencies of various microstrip antennas within a single neural model trained using parallel tabu search algorithm," *Electromagnetics*, vol. 25, no. 6, pp. 551–565, 2005.
- [35] Y. B. Tian, S. L. Zhang, and J. Y. Li, "Modeling resonant frequency of microstrip antenna based on neural network ensemble," *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, vol. 24, no. 1, pp. 78–88, 2011.
- [36] S. Han, Y. Tian, W. Ding, and P. Li, "Resonant frequency modeling of microstrip antenna based on deep kernel learning," *IEEE Access*, vol. 9, pp. 39067–39076, 2021.
- [37] J. Singh, A. P. Singh, and T. S. Kamal, "Design of circular microstrip antenna using artificial neural networks," *Proceedings of the World Congress on Engineering*, vol. 2, no. 1, pp. 9–12, London, UK, July 2011.
- [38] Ş. Sağıroğlu, K. Güney, and M. Erler, "Resonant frequency calculation for circular microstrip antennas using artificial neural networks," *International Journal of RF and Microwave Computer-Aided Engineering: Co-sponsored by the Center for Advanced Manufacturing and Packaging of Microwave, Optical, and Digital Electronics (CAMPmode) at the University of Colorado at Boulder*, vol. 8, no. 3, pp. 270–277, 1998.
- [39] W. Tian, D. Wu, Q. Chao, Z. Chen, and Y. Wang, "Application of genetic algorithm in $M \times N$ reconfigurable antenna array based on RF MEMS switches," *Modern Physics Letters B*, vol. 32, no. 30, p. 1850365, 2018.
- [40] G. Guang-Bin Huang, H. Hongming Zhou, X. Rui Zhang, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 2, pp. 513–529, 2012.
- [41] W. Ding, Y. Tian, S. Han, and H. Yuan, "Greedy broad learning system," *IEEE Access*, vol. 9, pp. 79307–79315, 2021.
- [42] C. L. Blake and C. J. Merz, "UCI repository of machine learning databases," 1998, <http://archive.ics.uci.edu/ml/datasets.php>.
- [43] J. S. Armstrong and F. Collopy, "Error measures for generalizing about forecasting methods: empirical comparisons," *International Journal of Forecasting*, vol. 8, no. 1, pp. 69–80, 1992.